

УДК 004.82:025.4

МНОГОУРОВНЕВОЕ СТРУКТУРИРОВАНИЕ ЗНАНИЙ И ФОРМИРОВАНИЕ ГИБКИХ КОНЦЕПТУАЛЬНЫХ АТЛАСОВ

Т.А. Гаврилова, О.Л. Малиновская

Аннотация

Структурирование знаний является ключевой фазой разработки интеллектуальных обучающих систем и систем дистанционного обучения. Описанная в работе методология проектирования интеллектуальных обучающих систем использует современную парадигму визуальной концептуализации. Рассмотрена также технология проектирования адаптивных обучающих систем, способных оперативно перестраивать сценарии обучения на основе использования модульных обучающих объектов и визуальных концептуальных графов или атласов.

Ключевые слова: инженерия знаний, учебные онтологии, интеллектуальные обучающие системы.

Введение

В последние годы в связи с широким развитием веб-технологий в сети Интернет доступно большое количество научно-познавательной информации и тематических обучающих ресурсов. Однако основная проблема состоит в том, что фрагменты информации, находящейся в сети, не связаны, и нередко дублируют друг друга. Это приводит к неэкономному и неэффективному использованию системных ресурсов (дискового объема сервера, трафика Интернета), а также к нерациональному расходованию времени пользователя-студента, осуществляющего поиск необходимой учебной информации в сети Интернет. Кроме того, исключительно важны технологии адаптации обучающих систем, позволяющие перестраивать учебный материал под потребности учащихся.

Таким образом, в настоящее время перспективной областью исследований является разработка дистанционных систем обучения, которые смогут предложить новые технологии и методы интерактивности и адаптивности и повысить интеллектуальность обычных обучающих систем.

Современный уровень развития технологий обучения существенным образом изменил представление об образовательном процессе. Разработка систем поддержки обучения в последние 20 лет становится ведущим трендом в педагогике всего мира [1–3]. Особенный интерес вызывают интеллектуальные системы, основанные на обработке знаний [4, 5]. В настоящей работе рассматривается актуальная проблема разработки современных интеллектуальных систем – технология проектирования адаптивных обучающих систем, способных оперативно перестраивать сценарии обучения на основе использования модульных обучающих объектов и визуальных концептуальных графов или атласов. Технология охватывает как сетевые системы дистанционного обучения, так и электронные учебники. В статье также описана методология структурирования учебного материала на независимые «учебные кубики» и формирования визуальных атласов, отражающих концептуальный скелет курса, способный гибко изменяться в соответствии с моделью

обучаемого. Модель обучаемого (или пользователя) включает различные социodemографические, психофизиологические и профессиональные параметры.

Результаты данного исследования внедрены в рамках российско-белорусского проекта «Проектирование Интеллектуальных Обучающих систем на основе модульных объектов и гибких концептуальных атЛАсов (ИОЛА)» (проект РФФИ). Они включают методологию и технологию, которые позволяют преподавателям разрабатывать гибкие сценарии обучения для интеллектуальных обучающих систем (ИОС) в различных областях науки и техники [6].

Создание ИОС условно можно разделить на две части:

- разработка методологии структурирования учебного материала на независимые «учебные кубики» и формирования визуальных атласов, отражающих концептуальный скелет курса;
- разработка архитектуры программного комплекса для проектирования интеллектуальных обучающих систем и технологии работы с системой.

В рамках настоящего исследования развивается концепция «обучающих объектов». Обучающий объект – это объект, содержащий как обучающую информацию по определенной тематике, так и служебную информацию в определенном стандартизированном формате.

Обучающие объекты можно повторно использовать и загружать в различные обучающие системы, которые по содержащейся в обучающем объекте информации могут легко распознать его содержимое и включить его в состав курса.

В проекте ИОЛА обучающие объекты и атласы представлены в виде онтологий. Последние годы ведущим методом структурирования и представления знаний в интеллектуальных системах являются онтологии, или иерархические концептуальные структуры. С методической точки зрения — это один из наиболее «систематических» и наглядных способов.

По определению Тома Грубера [7], впервые применившего это понятие вне философии, онтология – это уже не только философский термин, означающий учение о бытии. Термин переместился в область точных наук, где слабоформализованные концептуальные модели всегда сопутствовали математически строгим определениям. Все более широкое применение этот термин находит сейчас в «мягких» науках – менеджменте, экономике, электронном бизнесе, где структурирование информации представляет значительную сложность.

Понимание термина «онтология» зависит от контекста и целей его использования. В целом онтология, или концептуальная модель предметной области, состоит из иерархии понятий предметной области, связей между ними и законов, которые действуют в рамках этой модели. Онтология строится как сеть, состоящая из концептов и связей между ними. Связи могут быть различного типа, например, «является», «имеет свойство» и т. п. Концепты и связи имеют универсальный характер для некоторого класса понятий предметной области. Таким образом, под онтологией понимается спецификация концептуализации [7].

1. Задача структурирования материала

Необходимо отметить, что задача структурирования материала является неоднозначной, так как структурное представление сложной системы субъективно и различные авторы и преподаватели могут продуцировать разное видение системы. При разработке ИОС фиксируется лишь одна из возможных точка зрения на систему. Субъективность структуры зависит также от:

- целей структурирования информации;
- задач структурирования информации;
- целевой аудитории.

В общем случае процесс структурирования можно разбить на следующие этапы:

- постановка целей и задач структурирования;
- выделение аудитории и особенностей ее восприятия;
- определение приоритетов и стратегий структурирования;
- выбор названия и формы;
- разбиение всего массива информации на блоки;
- построение внутренней структуры блоков;
- соединение блоков в единую структуру.

В качестве модели представления информации использовались онтологии – один из наиболее перспективных и активно развивающихся подходов к представлению знаний [7–9]. Онтология определяет общий словарь для совместного использования информации в некоторой предметной области.

Как уже было сказано выше, под онтологией понимается некоторое формальное описание понятий, их свойств (или атрибутов) и отношений между ними в данной предметной области. В некоторых нотациях онтологий атрибуты называют слотами, а их значения – фасетами. Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний [10].

Использование онтологий обеспечивает следующие преимущества:

- возможность совместного использования людьми или программными агентами общего понимания структуры информации;
- возможность повторного использования знаний в предметной области;
- отделение декларативных знаний в предметной области от процедурных знаний;
- структурный анализ знаний в предметной области.

Вместо онтологий классического вида в нашем проекте использована система многоуровневых вложенных онтологий [5], в которой каждая онтология имеет три уровня (рис. 1):

- 1) фрагмент контента (Content fragment (CF) – онтологии нижнего уровня (в листьях содержатся конечные концепты, которые не могут быть раскрыты в онтологию, например текст, видео, аудио, изображение, таблица);
- 2) объект контента (Content Object (CO) – онтологии среднего уровня (содержат набор CF, CO и навигацию);
- 3) обучающий объект (Learning object (LO) – онтологии верхнего уровня (коллекции CO и связи между ними).

Онтологии верхнего уровня (или обучающие объекты – Learning objects) описываются по стандартам IEEE 1484.12.1 – 2002 (<http://www.imsglobal.org/metadata/>) и ADL SCORM Version 1.3. Использование концепции обучающих объектов позволяет рассматривать фрагменты знаний независимо друг от друга, расширять и модифицировать их, а также обеспечить к ним доступ другим обучающим системами. В то же время объекты не зависимы от программной платформы и контекста.

Для построения онтологий такого формата предлагается использовать методологию структурирования знаний, которая получила названия «Методология ИОЛА».

2. Методология инженерии знаний ИОЛА

Методология инженерии знаний ИОЛА основывается на широко известной и апробированной методике Protégé [11].

Она включает следующие «фундаментальные» правила разработки:

- 1) способ моделирования предметной области не является единственным; всегда существуют жизнеспособные альтернативы, а лучшее решение зависит от предполагаемого приложения и ожидаемых расширений (контекста);

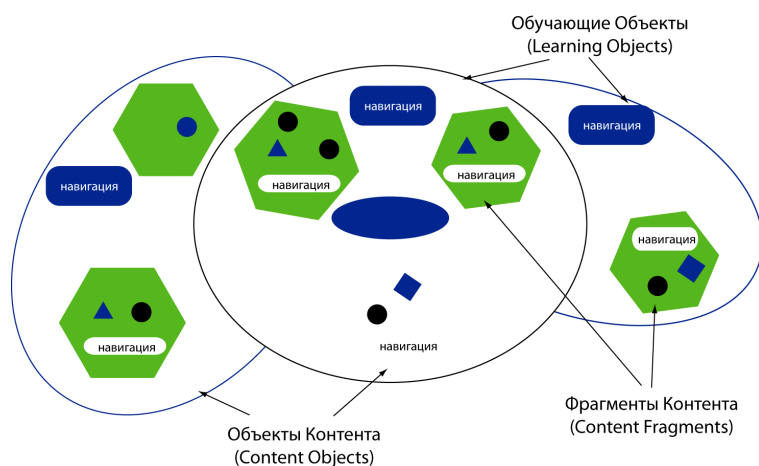


Рис. 1. Структура онтологий

- 2) онтология должна разрабатываться в несколько итераций;
- 3) понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в предметной области;
- 4) в качестве объектов онтологии должны использоваться существительные, описывающие предметную область;
- 5) в качестве отношений онтологии должны использоваться глаголы, описывающие связь объектов онтологий;
- 6) при построении онтологии необходимо понимать, для чего она будет использоваться и насколько детальной или общей она должна быть;
- 7) для начала необходимо построить не одну, а несколько альтернативных онтологий, из которых впоследствии будет выбрана одна, которая лучше других решит поставленную задачу и будет более наглядной, более расширяемой и более простой в обслуживании;
- 8) при построении онтологии необходимо помнить, что онтология – это модель реального мира и понятия в ней должны отражать эту реальность;
- 9) после определения единственной онтологии из множества альтернатив выбранную онтологию принимаем за начальную версию, которая будет подвержена итеративным изменениям;
- 10) в результате работы, как правило, должен произойти многократный пересмотр начальной онтологии;
- 11) процесс итеративного проектирования будет продолжаться в течение всего жизненного цикла онтологии.

Методология структурирования информации ИОЛА включает следующие шаги.

Шаг 1. Определение области и масштаба онтологии.

Для определения области и масштаба онтологий необходимо описать:

- какую область будет охватывать онтология;
- для чего она будет использоваться;
- на какие типы вопросов должна давать ответы информация в онтологии;
- кто будет использовать и поддерживать онтологию.

Шаг 2. Рассмотрение вариантов повторного использования существующих онтологий.

Всегда необходимо учитывать то, что было сделано ранее кем-то другим, и проверить, возможно ли улучшить и расширить существующие источники информации для конкретной предметной области и задачи.

В нашей методологии структурирования информации делается особый акцент на повторное использование существующих онтологий, на взаимодействие с другими приложениями, которые уже вошли в отдельные онтологии или контролируемые словари. В настоящее время множество онтологий уже доступно в электронном виде и может быть импортировано в используемую среду проектирования онтологий.

В литературе и сети Интернет существуют библиотеки повторно используемых онтологий, например библиотека онтологий Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>) или библиотека онтологий DAML (<http://www.darpa.mil/ixo/>). Существует также ряд общедоступных коммерческих онтологий (например, UNSPSC (www.unspsc.org), RosettaNet (www.rosettanet.org), DMOZ (www.dmoz.org)).

Для поддержки повторного использования информации, хранящейся в виде онтологий, мы предлагаем использовать международные стандарты метаописания данных в системах дистанционного обучения, такие как LOM (Learning Object Metadata) и SCORM (Sharable Content Object Reference Mode).

Шаг 3. Формирование глоссария, или перечисление основных терминов в онтологии.

Этот шаг подразумевает составление списка основных терминов предметной области с указанием иерархии понятий и основных свойств (атрибутов). Описываются также другие отношения (например, «часть – целое», «причина – следствие» и др.).

Шаг 4. Определение классов и иерархии классов.

Для разработки иерархии классов существует несколько возможных подходов:

- 1) *нисходящий* – процесс нисходящей разработки онтологии начинается с определения самых общих понятий предметной области с последующей конкретизацией понятий;
- 2) *восходящий* – процесс восходящей разработки онтологии начинается с определения самых конкретных классов, листьев иерархии с последующей группировкой этих классов в более общие понятия;
- 3) *комбинированный* – это сочетание нисходящего и восходящего подходов: сначала определяются более заметные понятия, а затем они соответствующим образом обобщаются и ограничиваются.

Все описанные здесь подходы являются равноправными, выбор подхода в большей степени зависит от личного взгляда на предметную область разработчика онтологий.

Шаг 5. Определение свойств классов – слотов.

После определения некоторого количества классов необходимо описать внутреннюю структуру понятий. С этой целью для каждого свойства из списка определяется, какой класс оно описывает. Эти свойства становятся слотами, привязанными к классам. В онтологии в качестве слотов могут быть использованы следующие типы свойств объектов:

- «внутренние» свойства;
- «внешние» свойства;
- части (если объект имеет структуру), они могут быть как физическими, так и абстрактными «частями»;
- отношения с другими индивидуальными концептами – это отношения между отдельными членами класса и другими элементами.

Все подклассы класса наследуют его слот. Слот должен быть привязан к самому общему классу, у которого может быть данное свойство.

Шаг 6. Определение фасетов слотов.

Слоты могут иметь различные фасеты, которые описывают

- тип значения;
- разрешенные значения;
- число значений (мощность) и другие свойства значений, которые могут принимать слот.

Мощность слота определяется количеством значений, которые может иметь слот. В некоторых системах различаются единичная мощность (возможно только одно значение) и множественная мощность (возможно любое число значений).

Другие системы позволяют определить минимальную и максимальную мощность для того, чтобы более точно описать количество значений слота.

Фасет *типа значения слота* описывает, какие типы значений можно ввести в слот. Вот список наиболее общих типов значений:

строка – самый простой тип значения, который используется в таких слотах, как название: значением является простая строка;

число (иногда используются более конкретные типы значений: Float (число с плавающей запятой) и Integer (целое число)) описывает слоты числовыми значениями;

булевы слоты – это простые флаги «да–нет»;

нумерованные слоты определяют список конкретных разрешенных значений слота;

слоты-экземпляры позволяют определить отношения между индивидуальными концептами. Слоты с типом значения Экземпляр также должны определять список разрешенных классов, экземпляры которых можно использовать.

Шаг 7. Создание экземпляров.

Последний шаг – это создание отдельных экземпляров классов в иерархии. Для определения отдельного экземпляра класса требуется:

- выбрать класс;
- создать отдельный экземпляр этого класса;
- ввести значения слотов.

3. Методология представления материала курса ИОЛА

Весь учебный материал курса предлагается разделить на следующие группы:

- исходные учебные данные (базис курса, представляющий из себя группу параграфов);
- ресурсы (HTML-страницы, изображения, таблицы, диаграммы, презентации, мультимедиа и др.).

Предлагается использовать следующие правила:

- материалы, представленные в базисе обучающего курса, должны быть структурированы в виде онтологий, а именно разбиты в соответствии со структурой обучающего курса;
- ресурсы обучающего курса подразделены по типам представляемых данных;
- учебные данные являются первостепенными и связаны с ресурсами путем ассоциации какого-либо ресурса с элементом учебного материала;
- набор конечных данных, предоставляемых пользователям обучающих курсов, содержит набор многоуровневых онтологий, описанных в соответствии со стандартами LOM и SCORM;
- материалы курса, структурированные в виде онтологий должны иметь метаописание на языке XML;
- формат XML-файла, содержащего метаописание обучающих объектов, строго определяется категориями и их атрибутами, заданными стандартом LOM;

- в файле метаописания находится один класс с именем LOM, девять одноименных подклассов с категориями модели LOM и некоторое количество атрибутов каждого подкласса,
- структура выходного XML-файла имеет следующий формат:

```
<LOM>
<название категории N>
<название атрибута N.m>
<название под-атрибута N.m.k>
.....
Значение
.....
</название податрибута N.m.k >
</название атрибута N.m >
</название категории N>
</LOM>
```

– визуализация онтологий может производиться одним из следующих средств:

- <http://protege.stanford.edu> – инструмент для создания, хранения и редактирования онтологий в OWL-формате;
- <http://hypergraph.sourceforge.net> – открытая легко настраиваемая платформа для работы с гиперболическими деревьями;
- технология touchgraph.

4. Архитектура программного комплекса ИОЛА

Программный инструмент ИОЛА предполагает возможность работы пользователя с разными типами информации (например, текст, HTML-страницы, изображения, таблицы, диаграммы, онтологии, мультимедиа и другие). При этом каждому типу ресурсов соответствует свой программный модуль, осуществляющий работу с данным ресурсом [12].

Так как система ИОЛА ориентирована на обучение, то первостепенным является исходный базис курса, оформленный средствами языка HTML. Контент курса структурирован традиционно по параграфам и расширен иллюстративным материалом в форме мультимедиа фрагментов.

Структура базиса курса является иерархической на всех уровнях. Учебные данные являются первостепенными и связаны с ресурсами путем ассоциации иллюстративных ресурсов с элементом учебного материала. Набор конечных данных, предоставляемых пользователям обучающих курсов, представляет собой набор многоуровневых онтологий, описанных в соответствии со стандартами LOM и SCORM.

Для воплощения в инструментари ИОЛА заданных требований была разработана архитектура, в состав которой входят четыре основных модуля:

- модуль управления информационными ресурсами (Resource Manager);
- модуль управления обучающими объектами (LOM Manager);
- модуль управления онтологиями (Ontology Manager);
- модуль адаптации (Adaptation Manager) – привязка групп и онтологий к стереотипной модели пользователя.

Полная архитектура системы представлена на рис. 2.

Рассмотрим каждый из модулей системы ИОЛА более подробно.

4.1. Модуль управления информационными ресурсами (Resource Manager). Загрузка исходных данных обучающего курса таких, как текст, HTML-страница, изображения различных форматов, презентации, аудио-, видео- и другие ресурсы, производится в модуль управления информационных ресурсов.

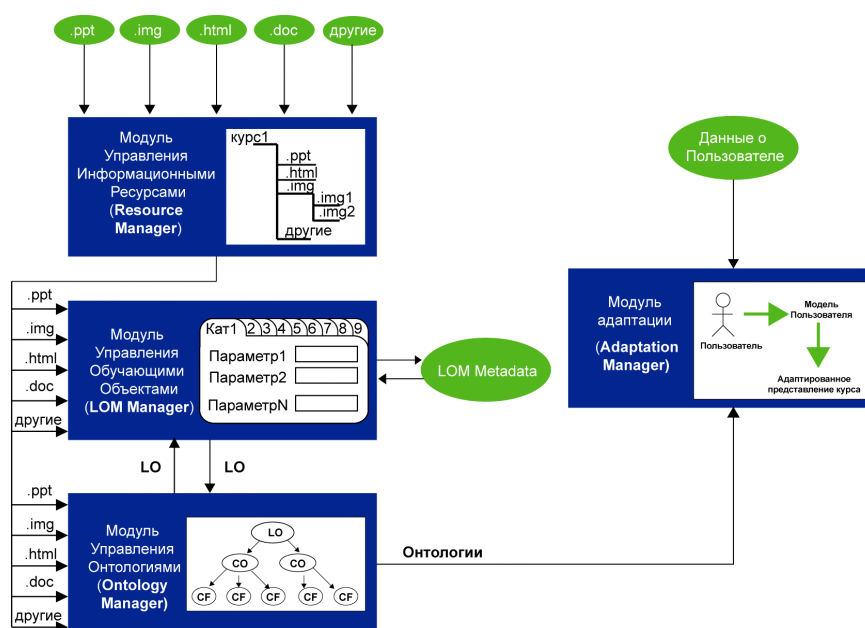


Рис. 2. Архитектура системы ИОЛА

Модуль управления (или менеджер) информационными ресурсами представляет древовидную структуру, где элементами верхнего уровня детализации являются разделы типов ресурсов, например изображения, текст, гипертекст, аудио, видео, а элементами нижнего уровня – сами обучающие ресурсы.

Система поддерживает строго древовидную форму представления информационных ресурсов в соответствии с иерархической структурой онтологий. Каждый элемент модуля менеджера является потомком базового класса, в котором определены базовые члены и методы. Для каждого типа информационных ресурсов установлены средства и методы просмотра, управления и редактирования данного элемента.

Помимо функции организации информационных ресурсов менеджер информационных ресурсов позволяет формировать группы («скелет» поля знаний обучающего курса).

Формирование базиса курса осуществляется при помощи добавления узлов (параграфов) к структуре курса и определения их параметров. Добавляемый элемент является потомком узла, к которому он добавляется. Каждый узел впоследствии может быть раскрыт как глава обучающего курса (HTML-текст), несущая конкретную справочную информацию.

Разработка групп позволяет реализовать модульную технологию обучения путем связывания определенной группы параграфов из сформированного ранее базиса, предоставляемых обучаемому, с конкретной моделью пользователя. Для определения этого набора необходимо использовать такие технологии, как сокрытие и аннотирование ссылок (параграфов) [13].

Разработчик курса может определять требования для фрагмента страницы и для страницы в целом. Требования являются булевскими выражениями над значениями концептов. Когда требования для фрагмента страницы принимают значение «истина», тогда фрагмент включается для отображения в курсе. В ином случае фрагмент исключается.

Разработчик также определяет для каждой группы значения набора оформительских параметров. Оформительские интерфейсные параметры, задаваемые так, как было описано выше, характеризуют отображение информации на экране и степень участия в этом пользователя. Среди них можно выделить [14]:

текущую оконную настройку. Веб-интерфейс систем дистанционного обучения можно оптимизировать под разрешение экрана (размер экранной области) пользователя;

расположение информации на экране. Расположение информации на экране влияет на усвоение материала пользователями в системах дистанционного обучения. Учет психологических, физиологических особенностей пользователя, его привычек позволит произвести адаптацию данного оформительского параметра веб-интерфейса;

вид меню. Задача адаптации данного параметра – создание меню, соответствующего психологическим характеристикам пользователя (в виде текста, графических изображений);

цвет фона;

цвет текста;

цвет гиперссылок.

Все эти параметры должны быть однозначно определяемы по модели пользователя и иметь возможность динамически изменяться. Таким образом, модель пользователя должна содержать не только список параграфов и частоту их посещения, но и список оформительских параметров с конкретными значениями и уникальный идентификатор группы, соответствующей данной модели пользователя.

Таким образом, условное описание модели пользователя принимает следующий вид:

$$M_j = \{group_id\},$$

group_id – идентификатор группы,

(1)

где $group_id = \{P_i, O_i\}$,

$$P_i = \bigcup p_i k_i, O_i = \bigcup F_i u_i,$$

F_i – оформительский параметр,

u_i – значение оформительского параметра F_i,

p_i – идентификатор параграфа,

k_i – булевский коэффициент (0 или 1),

P_i – группа,

O_i – оформление группы.

(2)

4.2. Модуль управления онтологиями (Ontology Manager). Модуль управления онтологиями предоставляет возможности создания и визуализации многоуровневых онтологий, содержащих три уровня вложенности, структура которых представлена на рис. 1.

Визуализация онтологий может производиться одним из следующих средств:

- <http://protege.stanford.edu> – инструмент для создания, хранения и редактирования онтологий в OWL-формате;
- <http://hypergraph.sourceforge.net> – открытая легко настраиваемая платформа для работы с гиперболическими деревьями;
- технология touchgraph.

Пример многоуровневой онтологии представлен на рис. 3.

Для поддержки сетевой технологии обучения необходимо к каждой модели пользователя привязать онтологию, по которой будет осуществляться навигация по обучающему курсу.

Пример Многоуровневых Вложенных Онтологий

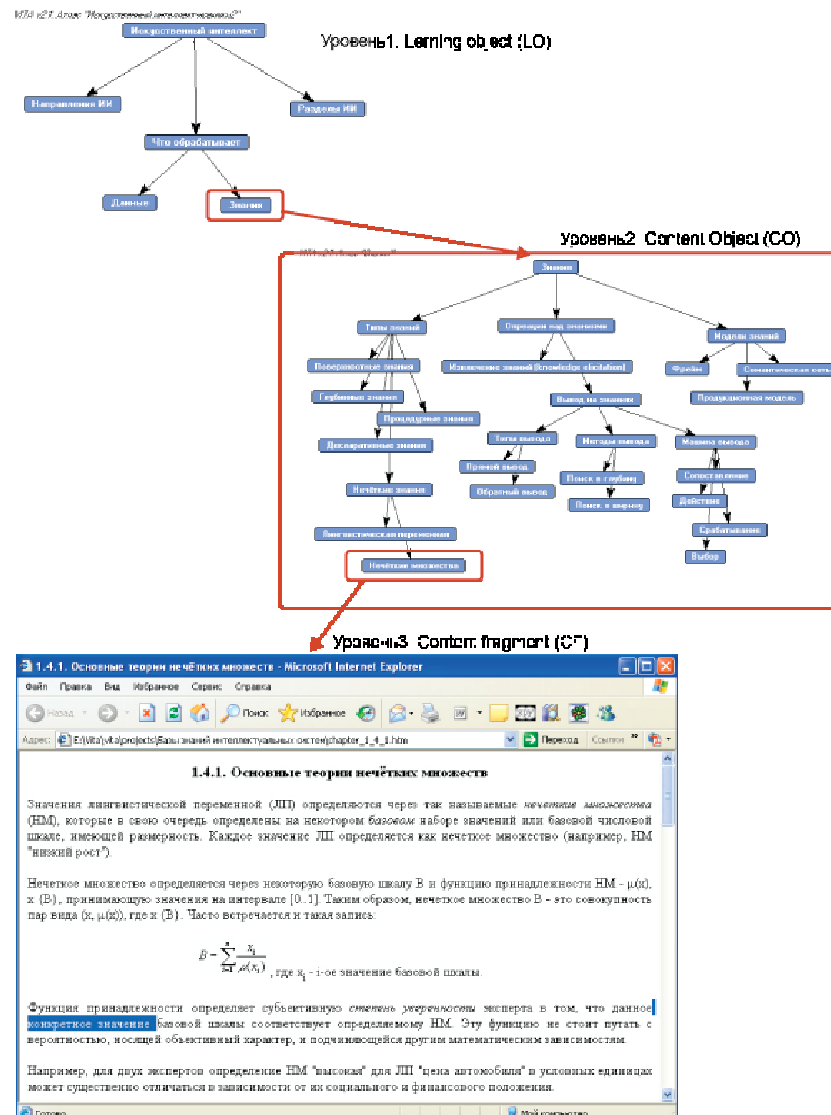


Рис. 3. Многоуровневая онтология понятия «Искусственный интеллект» и ее раскрытие на три уровня вложенности

Таким образом, условное описание модели пользователя будет расширено и примет следующий вид:

$$M_j = \{group_id, ontol_id\},$$

$$group_id - идентификатор_группы,$$

$$ontol_id - идентификатор_онтологии. \quad (3)$$

4.3. Модуль управления обучающими объектами (LOM Manager).
Модуль управления обучающими объектами LOM Manager используется для описания различных информационных ресурсов в соответствии с международными

стандартами LOM и SCORM, реализованного в виде заполнения фиксированного количества полей.

Основными функциями модуля являются:

- генерация метаописания информационных ресурсов;
- распознавание типа информационного ресурса в соответствии с его метаописанием.

Генерация метаописания информационных ресурсов происходит с помощью заполнения полей на форме и по нажатию клавиши происходит генерация XML-файла заданной структуры. Форма представляет собой окно с девятью закладками (по одной закладке на каждую категорию LOM).

Распознавание типа информационного ресурса происходит посредством чтения привязанного к ресурсу XML-файла.

4.4. Модуль Адаптации (Adaptation Manager). Для того чтобы разрабатываемая система была адаптивной, необходимо, чтобы она обеспечивала формирование и коррекцию модели пользователя.

Как уже отмечалось выше, существует две возможные методики адаптации: *стереотипная* и *индивидуальная* [14]. При индивидуальной технологии адаптации число моделей пользователя в системе неограничено. Каждому пользователю соответствует единственный уникальный профиль, что при большом числе пользователей системы, как в случае системы дистанционного обучения, может привести к неконтролируемому росту числа моделей пользователя и ухудшению функционирования системы. Поэтому в настоящей работе будем придерживаться стереотипной адаптации, когда в системе задано ограниченное, фиксированное число профилей, задаваемых разработчиком системы на этапе проектирования обучающего курса.

Модель пользователя представляется как вектор значений параметров, характеризующих данного пользователя [15]. Эти значения параметров динамически изменяются в зависимости от поведения обучаемого.

Один из параметров определяет уровень знаний пользователя в данной области курса, он косвенно может измеряться, например, количеством пройденных страниц.

4.5. Генератор теста на модель пользователя. Для удобной генерации теста на модель пользователя автором обучающего курса предлагается следующая схема:

- сначала осуществляется ввод количества моделей пользователя и происходит генерация уникальных идентификаторов моделей пользователя;
- для каждого идентификатора модели пользователя вводятся характеристики модели пользователя такие, как:
 - название модели пользователя,
 - описание модели пользователя,
 - идентификатора онтологии, соответствующей данной модели,
 - идентификатора группы, соответствующей данной модели,
 - значения выбранных оформительских параметров;
- далее происходит ввод списка вопросов теста на модель пользователя;
- для каждого вопроса теста осуществляется ввод списка ответов;
- в заключение происходит ввод таблицы-ключа, с помощью которой определяется по паре вопрос – ответ, к какой модели пользователя принадлежит тестируемый;
- вводятся понятия идентификатора модели пользователя Id_i и степень уверенности k_i , с которой тестируемый относится к модели пользователя Id_i . В качестве

коэффициентов k_i могут выступать логические переменные, а также может быть использована нечеткая логика, то есть коэффициенты k_i могут принимать значения в диапазоне от 0 до 1.

4.6. Организация работы в системе. Система предусматривает возможность выбора варианта дизайна из заранее подготовленного списка, а также возможность подгрузки дизайна в виде css-файлов.

Для сохранения обучающего курса имеются следующие режимы работы:

- публикация обучающего курса на веб-сервере;
- сохранение локальной версии обучающего курса для записи на CD-ROM;
- сохранение обучающего курса на Learning Management Server (домашний сервер).

4.7. Дополнительные функции. В систему планируется ввести следующие дополнительные инструменты:

- Assignment-tool – информационный ресурс (дает доступ к объяснению проблемы);
- Video assignment-tool – информационный видео-ресурс (дает доступ к объяснению проблемы);
- Calculator – калькулятор;
- Personal working space – настройки рабочего пространства, которые изменяются в процессе работы с обучающим курсом и сохраняются;
- Technical support tool – техническая помощь в работе с программой.

В некоторых источниках (см., например, [3]) рекомендуется вводить в обучающие системы специальный программный педагогический агент, помогающий в процессе обучения, например, объясняющий функции неизвестных инструментов, напоминающий о существовании неиспользуемых инструментов.

Заключение

Описанный в настоящей статье проект ИОЛА ориентирован на исследование актуальной проблемы – разработку технологии проектирования адаптивных обучающих систем, способных оперативно перестраивать сценарии обучения на основе использования модульных обучающих объектов и визуальных концептуальных графов или атласов.

Ядром технологии является идея структурирования учебного материала на независимые «учебные кубики» и формирования визуальных атласов, отражающих концептуальный скелет курса, способный гибко изменяться в соответствии с моделью обучаемого. В проекте ИОЛА обучающие объекты и атласы описываются в виде онтологий.

К сожалению, на сегодняшний день в России имеется лишь небольшое число научных коллективов, занимающихся разработками адаптивных систем. За рубежом существует много организаций, разрабатывающих стандарты для дистанционных обучающих систем, но каждая организация пропагандирует свой собственный стандарт. Предлагаемый подход структурирования и представления учебных материалов использует все лучшие наработки и международные стандарты в данной области и, кроме того, предлагает свой инновационный подход использования многоуровневого представления структуры знаний.

Работа частично поддержана РФФИ (проект № 08-07-00062).

Summary

T.A. Gavrilova, O.L. Malinovskaya. Multilevel Knowledge Structuring and Flexible Conceptual Atlases Design.

Knowledge structuring is a key phase in development of intelligent systems and distance learning. The described methodology of intelligent systems design uses a modern conceptual visualization paradigm. The technology of adaptive tutoring systems is also considered. These systems can promptly reconstruct learning scenarios on the basis of learning objects and visual conceptual graphs or atlases.

Key words: knowledge engineering, learning ontologies, intelligent tutoring systems.

Литература

1. *Брусиловский П.Л.* Адаптивные обучающие системы в World Wide Web: обзор имеющихся в распоряжении технологий. – URL: <http://ifets.ieee.org/russian/depository/WWWITS.html>.
2. *Собаева Е.В.* Интеллектуальные системы в дистанционном обучении. – 2003. – URL: <http://sumschool.sumdu.edu.ua/is-02/rus/lectures/sobaeva/sobaeva.htm>.
3. *Koedinger K.R., Corbett A.* Cognitive Tutors: Technology bringing learning science to the classroom // Sawyer K. The Cambridge Handbook of the Learning Sciences. – Cambridge: Cambridge Univ. Press, 2006. – P. 61–78.
4. *Пантелеев М.Г., Пузанков Д.В., Сазыкин П.В., Сергеев Д.А.* Интеллектуальные образовательные среды на основе технологий Семантического Web // Труды Междунар. конф. «Искусственные интеллектуальные системы». – М.: Изд-во физ.-мат. лит., 2002. – С. 236–243.
5. *Jovanović J., Gašević D., Verbert K., Duval E.* Ontology of Learning Object Content Structure // Proc. of the 2005 conf. on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology. – Amsterdam, The Netherlands: IOS Press, 2005. – P. 322–329.
6. *Geleverya T., Malinovskaya O., Gavrilova T., Golenkov V.* Prototyping Adaptive Online Learning Courses // Int. J. “Information Technologies and Knowledge”. – 2007. – V. 1, No 3. – P. 231–236.
7. *Gruber T.R.* A translation approach to portable ontology specifications // Knowledge Acquisition, – 1993. – V. 5, No 2. – P. 199–220.
8. *Гаврилова Т.А., Муромцев Д.И.* Интеллектуальные технологии в менеджменте: инструменты и системы. – СПб.: Изд. дом СПбГУ, Изд-во «Выш. шк. менеджмента», 2007. – 488 с.
9. *Митрофанова О.А., Константинова Н.С.* Онтологии как системы хранения знаний / Всерос. конкурсный отбор обз.-анал. статей по приоритетному направлению «Информационно-телекоммуникационные системы». – 2008. – 54 с. – URL: <http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf>.
10. *Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д.* Онтологии и тезаурусы. – Казань: Казан. гос. ун-т, 2006. – 197 с.
11. *Noy N.F., Deborah L. McGuinness* “Ontology Development 101: A Guide to Creating Your First Ontology”. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. – URL: http://protege.stanford.edu/publications/ontology_development/ontology101.html.
12. *Гелеверя Т., Малиновская О., Гаврилова Т.А., Курочкин М.А.* Система VITA-II для прототипирования учебных курсов on-line на основе онтологий // Proc. of Int. Conf. “Modern (e-) Learning”. – Sofia: FOI-COMMERCE, 2006. – P. 123–129.

13. *Брусиловский П.Л.* Технологии и методы адаптивной гипермедиа // User Modeling and User Adapted Interaction. – 1999. – V. 6, No 2–3. – P. 87–129. – URL: http://ifets.ieee.org/russian/depositary/Brusil_1996.zip.
14. *Васильева Е.И.* Некоторые подходы к формированию индивидуальных моделей обучаемых // Тез. докл. V Междунар. конф. «Современные технологии обучения» (СТО-99). – СПб., 1999. – С. 185–186.
15. *Гаврилова Т.А., Гелеверя Т.Е., Васильева Е.И.* Проблемы формирования модели пользователя в адаптивных системах дистанционного обучения // Материалы Междунар. науч.-метод. конф. «Дистанционное обучение – образовательная среда XXI века». – Минск, 2001. – С. 181–182.

Поступила в редакцию
13.04.09
Переработанный вариант
28.03.11

Гаврилова Татьяна Альбертовна – доктор технических наук, профессор, заведующий кафедрой информационных технологий в менеджменте Высшей школы менеджмента Санкт-Петербургского государственного университета.

E-mail: gavrilova@gsom.spb.ru

Малиновская Ольга Леонидовна – аспирант кафедры «Компьютерные интеллектуальные системы в проектировании» Санкт-Петербургского государственного политехнического университета.

E-mail: Olga.malinovskaya@gmail.com